

## ANALISIS SENTIMEN BERDASARKAN ULASAN PENGGUNA APLIKASI MYPERTAMINA PADA *GOOGLE PLAYSTORE* MENGUNAKAN METODE *NAÏVE BAYES*

Gilbert Darmawan<sup>1)</sup>, Syariful Alam<sup>2)</sup>, M. Imam Sulisty<sup>3)</sup>

<sup>1,2,3)</sup>Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana Purwakarta

<sup>1)</sup> gilbertdarmawan33@wastukencana.ac.id, <sup>2)</sup> syarifulalam@wastukencana.ac.id, <sup>3)</sup> imams@wastukencana.ac.id

---

### INFO ARTIKEL

#### Riwayat Artikel :

Diterima : 27 Juli 2023

Disetujui : 29 Juli 2023

#### Kata Kunci :

mypertamina, naïve bayes,  
sentimen analisis, google playstore

---

### ABSTRAK

Pada aplikasi *MyPertamina* peneliti menemukan banyak ulasan tidak sesuai dengan rating yang diberikan. Pada kasus ini juga peneliti membaca beberapa artikel / berita tentang pro dan kontra nya aplikasi *MyPertamina* dalam beberapa hal, Maka dari itu harus dilakukan analisis sentimen untuk mengetahui apakah ulasan pemakai aplikasi *MyPertamina* itu lebih banyak arah positif atau negatif. Penelitian ini menggunakan metode text mining yang terdiri dari tahapan *scraping data, labelling, cleaning, preprocessing (transformation, tokenization, filtering)*. Untuk algoritma yang digunakan yaitu *naïve bayes* karena mempunyai nilai probabilitas atau peluang tertinggi untuk pengklasifikasian data, untuk pembobotan menggunakan perhitungan TF-IDF, dan pengujian data menggunakan *confusion matrix* dan visualisasi menggunakan *wordcloud*. Hasil penelitian ini mengenai analisis sentimen ulasan pengguna *MyPertamina* pada *Google PlayStore* berjumlah 3948 data, disimpulkan bahwa ulasan pengguna *MyPertamina* tergolong negatif dengan hasil presentase 91% nilai akurasi, 92% nilai *precision*, dan 100% *recall* nya.

---

### ARTICLE INFO

#### Article History :

Received : July 27, 2023

Accepted : July 29, 2023

#### Keywords:

mypertamina, naïve bayes,  
sentiment analysis, google  
playstore

---

### ABSTRACT

*In the MyPertamina application, researchers found many reviews that did not match the rating given. In this case also researchers read several articles / news about the pros and cons of the MyPertamina application in several ways, therefore sentiment analysis must be done to find out whether the reviews of MyPertamina application users are more positive or negative directions. This research uses text mining method which consists of stages of data scraping, labelling, cleaning, preprocessing (transformation, tokenization, filtering). The algorithm used is naïve bayes because it has the highest probability value or opportunity for data classification, for weighting using TF-IDF calculation, and data testing using confusion matrix and visualization using wordcloud. The results of this study regarding sentiment analysis of MyPertamina user reviews on the Google Play Store totaling 3948 data, it can be concluded that MyPertamina user reviews are classified as negative with the percentage results of 91% accuracy value, 92% precision value, and 100% recall.*

## 1. PENDAHULUAN

Aplikasi *MyPertamina* merupakan salah satu lini bisnis PT. Pertamina (Persero) yang terus mengikuti perkembangan era globalisasi untuk tetap dapat bersaing dengan kompetitor lainnya. Aplikasi ini merupakan gabungan dari program loyalitas dan pembayaran elektronik yang PT dapat memberikan pengalaman para pengguna PT Pertamina. Aplikasi ini dapat memudahkan transaksi konsumen, Saat ini *MyPertamina* telah diunduh lebih dari 5 juta pengguna dengan rating 2,9 dan lebih dari 200.000 ulasan (Indrayanto, Ratnawati and Rahayudi, 2023).

Pada Aplikasi *MyPertamina*, peneliti menemukan banyaknya rating yang tidak sesuai dengan rating yang diberikan, maka dari itu peneliti memutuskan untuk melakukan analisis sentimen. Analisis sentimen merupakan proses memahami, mengekstrak, dan juga mengolah suatu data informasi teks secara otomatis untuk mendapatkan informasi sentimen yang terdapat pada suatu kalimat (Aputra *et al.*, 2019). Oleh karena itu, analisis opini harus dilakukan untuk mengetahui *review* pengguna aplikasi *MyPertamina* lebih banyak ke arah positif atau negatif.

Terdapat beberapa algoritma yang dapat digunakan untuk melakukan analisis sentimen seperti algoritma *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Naïve Bayes*. Berdasarkan beberapa referensi artikel yang sudah peneliti baca, algoritma *Naïve Bayes* memiliki hasil akurasi yang cukup tinggi dan segi waktu yang terbilang cepat dalam melakukan analisis sentimen dibandingkan algoritma klasifikasi yang lain.

Pada penelitian sebelumnya oleh (Fitriana, Utami and Al Fatta, 2021) algoritma SVM memiliki tingkat akurasi yang lebih tinggi sebesar 90,47%, 90,23% untuk nilai presisi nya, dan 90,78% untuk nilai *recall* nya dengan kecepatan waktu 11 detik. Pada algoritma *Naïve Bayes* mendapatkan nilai akurasi sebesar 88,64%, 87,32% untuk nilai presisi nya, dan 88,13% untuk nilai *recall* nya dengan kecepatan waktu yang lebih baik yaitu 8,1 detik. Adapun pada penelitian terdahulu yang dikemukakan oleh (Putri, Ratnawati and Brata, 2023), Pada penelitian tersebut dari ke dua algoritma yang dipakai didapatkan hasil sebagai berikut untuk

algoritma *Naïve Bayes* sebesar 88,5% untuk nilai akurasi, 88,7% untuk nilai presisi, dan 88,2% untuk nilai *recall* nya. Sedangkan untuk algoritma *K-Nearest Neighbor* (KNN) didapatkan hasil 84,8% untuk nilai akurasi, 85,4% untuk nilai presisi, dan 84,6% untuk nilai *recall* nya. Pada penelitian serupa oleh (Astari, Dewa Gede Hendra Divayana and Gede Indrawan, 2020), Setelah setiap proses pelabelan selesai dilakukan menunjukkan bahwa algoritma *Naïve Bayes* mendapatkan 89% untuk nilai akurasi, 83% untuk nilai presisi, dan 87% untuk nilai *recall* nya. Hasil tersebut didapat dari hasil data uji sebanyak 65 data, terdapat 20 perkiraan data rating positif, 4 data rating positif memprediksi review negatif, 3 data review negatif memprediksi review positif, dan 38 data review negatif memprediksi dengan tepat.

Algoritma yang akan digunakan adalah algoritma *Naïve Bayes* berdasarkan penelitian-penelitian terdahulu yang sudah disebutkan diatas, Algoritma *Naïve Bayes* dipilih karena memiliki keunggulan yaitu jumlah data *training* yang tergolong sedikit untuk memprediksi parameter yang diperlukan pada proses klasifikasi nya. algoritma *Naïve Bayes* juga terbukti memiliki kecepatan dan akurasi yang cukup tinggi saat diaplikasikan dalam *dataset* dengan jumlah yang besar (Fauzan and Hikmah, 2022). Analisis sentimen bertujuan untuk mengetahui sentimen pengguna aplikasi *MyPertamina* berdasarkan ulasan pada *Google PlayStore* dapat diterima atau tidak oleh masyarakat menggunakan algoritma *Naïve Bayes*.

Pada penelitian ini, fokus penelitian ditujukan pada analisis sentimen pengguna aplikasi *MyPertamina* berdasarkan ulasan pada *Google PlayStore*. Penelitian ini akan menjelaskan secara jelas proses-proses yang akan menghasilkan analisis sentimen dengan menerapkan algoritma *Naïve Bayes* dengan menggunakan bahasa pemrograman *Python*. Algoritma *Naïve Bayes* akan digunakan sebagai metode untuk menghasilkan prediksi terhadap suatu ulasan dan membagi nya kedalam 2 kategori yaitu positif dan negatif. Algoritma *Naïve Bayes* ini dipilih dan digunakan karena sudah banyak digunakan untuk proses *text mining* karena memiliki nilai akurasi yang tinggi dan juga memiliki waktu pengklasifikasian yang

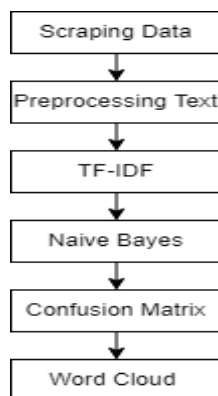
tergolong cepat dalam analisis sentimen pada penelitian-penelitian terdahulu (Ama *et al.*, 2022).

Berdasarkan latar belakang yang sudah dipaparkan diatas, peneliti akan melakukan analisis sentimen terhadap aplikasi MyPertamina berdasarkan ulasan pada Google PlayStore, yang diharapkan dapat berguna dan bermanfaat bagi PT. Pertamina (Persero) untuk memperbaiki kekurangan yang dikeluhkan oleh masyarakat dan meningkatkan kualitas produk nya menjadi lebih baik dan bermanfaat bagi masyarakat pengguna aplikasi MyPertamina.

## 2. METODE

### 2.1. Kerangka Penelitian

Pada penelitian ini, terdapat beberapa proses atau tahapan yang harus dilakukan untuk mendapatkan hasil yang terbaik. Proses yang harus dilakukan yaitu terdiri dari *scraping data* atau pengumpulan data, lalu *preprocessing text*, lalu data akan dilakukan pembobotan kata menggunakan proses TF-IDF, setelah itu data akan dikelompokkan menggunakan algoritma *Naive Bayes*, lalu untuk tahap evaluasi data menggunakan *Confusion Matrix* dan untuk tahap visualisasi menggunakan *WordCloud* yang dapat dilihat pada gambar 1 dibawah ini.



Gambar 1. Kerangka kerja penelitian

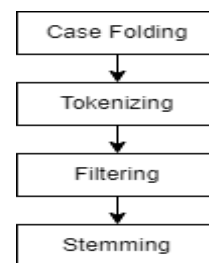
### 2.2. Scraping Data

Pada tahap ini, pengumpulan data atau *scraping data* dilakukan menggunakan Bahasa pemrograman Python menggunakan tools Google Colaboratory dan package google-play-scraper. Pada *scraping data* ini peneliti mengambil beberapa data yang dibutuhkan seperti username, komentar, rating, jumlah like,

dan sebagainya dan menyimpan data nya dalam bentuk file .csv.

### 2.3. Preprocessing Text

Tahap selanjutnya adalah *preprocessing text*, ditahap ini data yang sudah di *scraping* akan di sama ratakan bentuk dan formatnya agar data dapat diproses pada tahap selanjutnya. Tahap *preprocessing text* ini terdiri dari 4 tahapan yaitu *case folding*, *tokenizing*, *filtering*, dan *stemming* yang dapat dilihat pada gambar 2 dibawah ini



Gambar 2 Tahap *preprocessing text*

#### 1. Case Folding

Tahap *case folding* ini merupakan proses penyamarataan huruf yang sebelumnya terdapat huruf kapital diubah menjadi huruf kecil semua.

#### 2. Tokenizing

Tahap selanjutnya adalah tahap *tokenizing*, tahapan ini merupakan proses pemotongan atau pemisahan kalimat menjadi kata per kata.

#### 3. Filtering

Tahap selanjutnya setelah *tokenizing* adalah tahap *filtering*, tahap menghilangkan atau menghapus kata-kata yang tidak dibutuhkan pada proses berikutnya.

#### 4. Stemming

Tahapan selanjutnya yaitu tahap *stemming* yaitu untuk mengubah kata yang memiliki imbuhan menjadi kata dasar.

### 2.4. Term Frequency Inverse Document Frequency (TF-IDF)

Setelah dilakukan tahap *preprocessing text*, tahapan selanjutnya yang harus dilakukan adalah tahap TF-IDF. Tahap ini melakukan proses *tf* (term weighting) yang setelahnya akan diberikan bobot setiap kata yang dimana bobot tersebut mengindikasikan pentingnya sebuah term pada sebuah dokumen. Perhitungan bobot tiap term dicari keseluruhan pada setiap dokumen bertujuan untuk mengetahui dan mendapat

jumlah kata yang sering muncul pada dokumen dan semakin sering kata tersebut muncul maka semakin tinggi pula bobot atau nilai term tersebut.

## 2.5. Naïve Bayes

Tahapan setelah TF-IDF yaitu tahapan klasifikasi model menggunakan algoritma *Naïve Bayes*. *Naïve Bayes* merupakan metode klasifikasi sederhana yang bisa menghitung semua kemungkinan dengan cara menggabungkan jumlah kombinasi dan kerapatan nilai dari suatu dataset yang sudah diperoleh (Rachman and Handayani, 2021). Algoritma klasifikasi *naïve bayes* merupakan model algoritma yang sudah sering digunakan untuk melakukan klasifikasi data dengan menggunakan perhitungan probabilitas. Perhitungan probabilitas untuk setiap kata pada dokumen atau dataset dapat menghasilkan 2 klasifikasi yaitu sentimen positif dan negatif dari proses komputasi yang sebelumnya pada pembobotan kata TF-IDF (Chatrina, Siregar, Ruli, A, Siregar and Yoga, Distra, Sudirman, 2020). Adapun kelebihan menggunakan metode ini yaitu membutuhkan sedikit data training untuk menentukan estimasi parameter yang diperlukan dalam proses klasifikasi. Karena diasumsikan sebagai variabel yang bebas (Alam, Resmi and Masripah, 2022).

## 2.6. Confusion Matrix

Tahap selanjutnya ialah tahap evaluasi menggunakan *confusion matrix*, confusion matrix adalah metode evaluasi yang digunakan untuk mengukur kinerja metode klasifikasi. Selama pengukuran efisiensi terdapat 4 istilah sebagai representasi hasil dari proses klasifikasi. Keempat istilah itu antara lain adalah *True Positive* (TF), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). (Karsito and Susanti, 2019). Setelah didapatkan nilai dari keempat istilah tersebut maka setelah itu dapat didapatkan nilai untuk akurasi, presisi, dan recall nya pada rumus berikut :

$$accuracy = \frac{TP+TN}{Total} (1)$$

Keterangan :

TP = hasil data dengan nilai prediksi positif dan nilai aktual positif

TN = hasil data dengan nilai prediksi negatif dan nilai negatif negatif

$$presisi = \frac{TP}{TP+FP} (2)$$

Keterangan :

TP = hasil data dengan nilai prediksi positif dan nilai aktual positif

FP = hasil data dengan nilai prediksi positif dan nilai aktual negatif

$$recall = \frac{TP}{TP+FN} (3)$$

Keterangan :

TP = hasil data dengan nilai prediksi positif dan nilai aktual positif

FN = hasil data dengan nilai prediksi negatif dan nilai sebenarnya positif. Sumber (Normawati and Prayogi, 2021)

## 2.7. WordCloud

Tahap selanjutnya setelah dilakukan evaluasi yaitu tahap visualisasi menggunakan word cloud. Tahap ini digambarkan dengan word cloud pada seluruh sentimen keseluruhan seluruh dokumen ulasan positif dan negatif tentang aplikasi MyPertamina.

## 3. HASIL DAN PEMBAHASAN

### 3.1. Hasil Scraping Data

Pengambilan data dilakukan dengan metode scraping menggunakan *Google Colaboratory* dengan menginstall *google-play-scraper* untuk mendapatkan data ulasan pada *google playstore*. Data yang didapatkan berupa *username*, ulasan, *rating*, jumlah like, waktu dan tanggal, dan lainnya sebanyak 5000 data dan setelah di *cleaning* menjadi 3948 data seperti pada gambar 3 dibawah

```
from google_play_scraper import app
import pandas as pd
import numpy as np

[ ] from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.tencent.qqlive118n',
    lang='id',
    country='id',
    sort=Sort.MOST_RELEVANT,
    count=5000,
    filter_score_with=None
)
```

Gambar 3. Proses scraping data

### 3.2. Hasil Preprocessing Text

Preprocessing text merupakan tahapan selanjutnya setelah tahap scraping, pada tahap ini data akan di samakan seluruh bentuk dan format nya agar pada tahap selanjutnya data dapat diproses dan diolah. Tahap preprocessing text terdiri dari 4 tahapan yaitu, *case folding*, *tokenizing*, *filtering*, dan *stemming*.

Tabel 1 Hasil tahap *case folding*

Sebelum <i>case folding</i>	Sesudah <i>case folding</i>
Sangat membantu & memberi kemudahan bagi rakyat kecil	sangat membantu & memberi kemudahan bagi rakyat kecil

Tabel 1 menunjukkan hasil dari proses case folding dimana data sebelum di case folding masih terdapat huruf kapital lalu setelah proses case folding semua huruf kapital diubah menjadi huruf kecil semua.

Tabel 2 Hasil tahap tokenizing

sangat	membantu	&	memberi
kemudahan	bagi	rakyat	kecil

Setelah proses *case folding*, data akan dilakukan proses tokenizing yaitu memecah kalimat ulasan menjadi kata per kata seperti yang dapat dilihat pada tabel 2.

Tabel 3 Hasil tahap filtering

Sebelum filtering	Sesudah filtering
sangat membantu & memberi kemudahan bagi rakyat kecil	sangat membantu memberi kemudahan bagi rakyat

Setelah tahap tokenizing data akan masuk kedalam tahap filtering yaitu memfilter atau menghilangkan kata-kata yang tidak diperlukan dalam tahapan berikutnya seperti yang dapat dilihat pada tabel 3.

```
[ ] !pip -q install sastrawi
----- 209.7/209.7 kB 13.6 MB/s eta 0:00:00
```

Gambar 4 Proses install sastrawi

Tahapan selanjutnya adalah proses stemming, yaitu data yang sudah diproses pada tahap sebelumnya akan diubah kata imbuhan nya menjadi kata dasar. Untuk proses stemming ini diperlukan library Sastrawi untuk proses selanjutnya dilakukan. Untuk hasil proses stemming dapat dilihat pada tabel 4.

Tabel 4 Hasil tahap *stemming*

Sebelum	Sesudah
sangat membantu memberi kemudahan bagi rakyat	sangat bantu beri mudah bagi rakyat

### 3.3. Hasil TF-IDF

Setelah proses preprocessing selesai dilakukan tahapan selanjutnya yaitu dilakukan *term weighting*, yang akan diberi bobot (nilai). Pada proses ini yang pertama dilakukan yaitu *term frequency*, *inverse document frequency*, dan tahap terakhir yaitu *term frequency inverse document frequency*. Berikut adalah contoh sampel data yang akan digunakan yang akan ditandai sebagai 5 sentimen positif dan 5 sentimen negatif. Data tersebut digunakan sebagai data training yang dapat dilihat pada tabel 5.

Tabel 5 Sampel data

Teks	Sentimen
aplikasi bagus data masuk barcode muncul	Positif
aplikasi bagus mudah transaksi nontunai	Positif
inovasi aplikasi bagus mudah bayar uang fisik cashless guna mudah tampil fitur tarik	Positif
aplikasi gampang interfacenya userfriendly	Positif
aplikasi mudah bayar via cashless tinggal	Positif

Teks	Sentimen
scan langsung bayar cepat antri bayar pakai cash banget mbak mbak cari kembali pakai aplikasi satset selesai pokok bagus aplikasi	
aplikasi susah buka repot beli solar bayar ribet banget	Negatif
aplikasi ngebug daftar susah alih aplikasi manual aplikasi tampung juta konsumen daftar aplikasi dukung bijak perintah bijak paksa	Negatif
aplikasi buruk login bug	Negatif
aplikasi ribet sejarah playstore	Negatif
login aplikasi susah ribet isi bbm	Negatif

Setelah mendapatkan sampel data, lalu peneliti mencari nilai term frequency dengan menulis angka 1 untuk kata yang muncul dan angka 0 untuk kata yang tidak muncul pada setiap dokumen yang dapat dilihat pada tabel 6.

Tabel 6 Nilai TF

term	TF			
	D1	D2	...	D10
aplikasi	1	1	...	1
bagus	1	1	...	0
cashless	0	0	...	0
mudah	0	1	...	0
susah	0	0	...	1
ribet	0	0	...	1
bug	0	0	...	0
bayar	0	0	...	0

Setelah didapatkan hasil term frequency tahapan selanjutnya yaitu menghitung nilai document frequency yang didapat dari berapa

banyak kata yang ada dalam setiap dokumen. Kemudian total dari seluruh dokumen dibagi dengan document frequency untuk menghasilkan nilai D/df yang dapat dilihat pada gambar 7

term	TF										df	D/df	idf (log D/df)	1+idf
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10				
aplikasi	1	1	1	1	1	1	1	1	1	1	10	1	0	1
bagus	1	1	1	0	1	0	0	0	0	0	4	2.5	0.397940009	1.39794
cashless	0	0	1	0	1	0	0	0	0	0	2	5	0.698970004	1.69897
mudah	0	1	1	0	1	0	0	0	0	0	3	3.333	0.522878745	1.522879
susah	0	0	0	0	0	1	1	0	0	1	3	3.333	0.522878745	1.522879
ribet	0	0	0	0	1	0	0	1	1	1	3	3.333	0.522878745	1.522879
bug	0	0	0	0	0	1	1	0	0	0	2	5	0.698970004	1.69897
bayar	0	0	1	0	1	1	0	0	0	0	3	3.333	0.522878745	1.522879

Gambar 7 Hasil inverse document frequency

Keterangan :

df : jumlah kemunculan kata pada setiap dokumen

D/df : jumlah total dokumen dibagi dengan jumlah kemunculan kata.

Log idf : jumlah perhitungan logaritma D/df

1+idf : hasil dari perhitungan nilai dari log idf ditambah 1, dimana angka 1 merupakan nilai inverse.

Tahapan selanjutnya ialah tahap term frequency inverse document frequency untuk mendapatkan nilai tf-idf yaitu dengan cara mengalikan bobot tiap kata dari seluruh dokumen. Untuk jumlah perhitungan  $W=tf(idf+1)$  yaitu jumlah perhitungan dari tiap kemunculan kata yang dikalikan nilai dari 1+idf. Untuk hasil perhitungan tf-idf dapat dilihat pada gambar 8.

term	W=tf(idf+1)									
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
aplikasi	1	1	1	1	1	1	1	1	1	1
bagus	1.397940009	1.39794	1.39794	0.0000	1.39794	0.0000	0.0000	0.0000	0.0000	0.0000
cashless	0.0000	0.0000	1.69897	0.0000	1.69897	0.0000	0.0000	0.0000	0.0000	0.0000
mudah	0.0000	1.52288	1.52288	0.0000	1.522879	0.0000	0.0000	0.0000	0.0000	0.0000
susah	0.0000	0.0000	0.0000	0.0000	0.0000	1.52288	1.522879	0.0000	0.0000	1.52288
ribet	0.00000	0.0000	0.0000	0.0000	0.0000	1.52288	0.0000	0.0000	1.52288	1.52288
bug	0.00000	0.0000	0.0000	0.0000	0.0000	0.0000	1.69897	1.69897	0.0000	0.0000
bayar	0.00000	0.0000	1.52288	0.0000	1.522879	1.52288	0.0000	0.0000	0.0000	0.0000
	2.397940009	3.92082	7.14267	1	7.142668	5.56864	4.221849	2.69897	2.52288	4.04576

Gambar 8 hasil perhitungan tf-idf

### 3.4. Hasil Klasifikasi Naïve Bayes

Setelah proses tf-idf selesai dilakukan, maka didapatkan data yang telah siap untuk diolah untuk melakukan pengujian model Naïve Bayes pada python lalu membagi data menjadi data training dan data testing dengan perbandingan 80 : 20. Setelah didapatkan data training, untuk pengujian model dapat dilihat pada gambar 9

```
#Training Model
from datetime import datetime
start_time = datetime.now()
from joblib import dump
#algoritme fitting

text_algorithm = MultinomialNB()

model = text_algorithm.fit(x_train, y_train)

# save the model to disk
dump(model, filename="model_sentiment_naive.joblib")

end_time = datetime.now()
result_time = end_time-start_time
print("Duration:",result_time)
```

Duration: 0:00:00.055629

Gambar 9 Pengujian model *naïve bayes*

Setelah melakukan pengujian maka didapatkan waktu 5 detik untuk pengujian nya yang bisa dikategorikan algoritma *naïve bayes* cukup baik dalam waktu pengujiannya.

### 3.5. Hasil Confusion Matrix

Tahap selanjutnya yaitu tahap evaluasi menggunakan confusion matrix. Tahapan evaluasi ini bertujuan untuk mengecek kebenaran hasil klasifikasi data testing seperti pada tabel tabel 7.

Tabel 7 Hasil *Confusion Matrix*

	True	False
Negatif	3	0
Positif	721	66

Setelah didapatkan nilai untuk *True Positive* (TF), *True Negative* (TN), *False Postive* (FP), dan *False Negative* (FN), tahap selanjutnya adalah melakukan pengujian pada python dengan rumus seperti yang disebutkan sebelumnya, lalu didapatkan hasil sebagai berikut yaitu nilai akurasi 91,6 %, presisi 91,6 %, dan *recall* nya 100 %.

### 3.6. Hasil WordCloud

Tahapan terakhir yaitu adalah tahap visualisasi data menggunakan *wordcloud* untuk mengetahui kata yang sering muncul pada setiap sentimen. Berikut adalah hasil *wordcloud* pada sentimen positif yang dapat dilihat pada gambar 10.



Gambar 10 WordCloud sentimen positif

Pada gambar 10 dapat dilihat kata yang sering muncul pada sentimen positif yaitu sebagai berikut : “aplikasi”, “pertamina”, “bayar”, “mudah”, “bagus” dan lain lain. Untuk word cloud sentimen negatif dapat dilihat pada gambar 11.



Gambar 11 WordCloud sentimen negatif

Pada gambar 11 dapat dilihat kata yang sering muncul pada sentimen negatif yaitu sebagai berikut : “aplikasi”, “daftar”, “ribet”, “error”, “susah” dan lain lain.

## 4. PENUTUP

### 4.1. Kesimpulan

Dari hasil penelitian diatas yakni mengenai analisis sentiment ulasan pengguna *MyPertamina* pada *Google PlayStore* yang berjumlah 3948 data dan sudah melewati tahap *preprocess text* yang terdiri dari *transformation*, *tokenization*, *filtering*, *stemming* dan klasifikasi menggunakan algoritma *Naïve Bayes* serta evaluasi data menggunakan *Confusion Matrix* menggunakan *Google Colaboratory* dapat disimpulkan bahwa tanggapan / ulasan pengguna *MyPertamina* tergolong negatif dengan hasil presentase 91,6% untuk nilai akurasi (*accuracy*), 92% untuk nilai *precision*, dan 100% untuk

tingkat keberhasilan *recall* nya. Nilai ini menunjukkan bahwa klasifikasi algoritma *Naive Bayes* dinilai cukup baik dalam pemrosesan data ulasan *play store*, ini dikarenakan hasil presentase akurasi nya mencapai 91%. Berdasarkan nilai tersebut membuktikan bahwa sentiment masyarakat mengenai aplikasi *MyPertamina* pada *Google PlayStore* tergolong negatif.

#### 4.2. Saran

- a. Untuk Peneliti selanjtnya bisa mencoba atau menggunakan algoritma yang lainnya dalam menganalisis sentiment seperti algoritma *K-Nearest Neighbor*, *SVM*, ataupun algoritma lainnya.
- b. Dapat memperluas dataset yang lebih besar dan lebih beragam agar dapat membantu meningkatkan kinerja model analisis sentiment. Melibatkan data dari berbagai sumber, industri, atau Bahasa agar dapat memberikan wawasan yang lebih luas tentang sentimen

#### 5. DAFTAR PUSTAKA

- Aaputra, S.A. *et al.* (2019) ‘Sentiment Analysis Analysis of E-Wallet Sentiments on Google Play Using the Naive Bayes Algorithm Based on Particle Swarm Optimization’, *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 3(3), pp. 377–382. Available at: <https://doi.org/10.29207/resti.v3i3.1118>.
- Alam, S., Resmi, M.G. and Masripah, N. (2022) ‘Classification of Covid-19 vaccine data screening with Naive Bayes algorithm using Knowledge Discovery in database method’, *Journal of Computer Networks, Architecture and High Performance Computing*, 4(2), pp. 177–185. Available at: <https://doi.org/10.47709/cnahpc.v4i2.1584>.
- Ama, A.U.T. *et al.* (2022) ‘Analisis Sentimen Customer Feedback Tokopedia Menggunakan Algoritma Naive Bayes’, *Jurnal Sistem Komputer dan Informatika (JSON)*, 4(1), p. 50. Available at: <https://doi.org/10.30865/json.v4i1.4783>.
- Astari, N.M.A.J., Dewa Gede Hendra Divayana and Gede Indrawan (2020) ‘Analisis Sentimen Dokumen Twitter Mengenai Dampak Virus Corona Menggunakan Metode Naive Bayes Classifier’, *Jurnal Sistem dan Informatika (JSI)*, 15(1), pp. 27–29. Available at: <https://doi.org/10.30864/jsi.v15i1.332>.
- Chatrina, Siregar, N., Ruli, A, Siregar, R. and Yoga, Distra, Sudirman, M. (2020) ‘Implementasi Metode Naive Bayes Classifier (NBC) Pada Komentar Warga Sekolah Mengenai Pelaksanaan Pembelajaran Jarak Jauh (PJJ)’, *Jurnal Teknologia*, 34(1), pp. 102–110. Available at: <https://aperti.ejournal.id/teknologia/article/view/67>.
- Fauzan, A.C. and Hikmah, K. (2022) ‘Implementasi Algoritma Naive Bayes Dalam Analisis Polarisasi Opini Masyarakat Terkait Vaksin Covid-19’, *Rabit: Jurnal Teknologi dan Sistem Informasi Univrab*, 7(2), pp. 122–128. Available at: <https://doi.org/10.36341/rabit.v7i2.2403>.
- Fitriana, F., Utami, E. and Al Fatta, H. (2021) ‘Analisis Sentimen Opini Terhadap Vaksin Covid - 19 pada Media Sosial Twitter Menggunakan Support Vector Machine dan Naive Bayes’, *Jurnal Komtika (Komputasi dan Informatika)*, 5(1), pp. 19–25. Available at: <https://doi.org/10.31603/komtika.v5i1.5185>.
- Indrayanto, C.G., Ratnawati, D.E. and Rahayudi, B. (2023) ‘Analisis Sentimen Data Ulasan Pengguna Aplikasi MyPertamina di Indonesia pada Google Play Store menggunakan Metode Random Forest’, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7(3), pp. 1131–1139. Available at: <http://j-ptiik.ub.ac.id>.
- Karsito and Susanti, S. (2019) ‘Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma Naive Bayes Di Perumahan Azzura Residencia’, *Jurnal Teknologi Pelita Bangsa*, 9, pp. 43–48.
- Normawati, D. and Prayogi, S.A. (2021) ‘Implementasi Naive Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter’, *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 5(2), pp. 697–711.



- Putri, R.T.S.A., Ratnawati, D.E. and Brata, D.W. (2023) ‘Perbandingan Naïve Bayes dan K-Nearest Neighbor untuk Analisis Sentimen Aplikasi Gapura UB Berdasarkan Ulasan Pengguna pada Playstore’, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7(1), pp. 229–236. Available at: <http://j-ptiik.ub.ac.id>.
- Rachman, R. and Handayani, R.N. (2021) ‘Klasifikasi Algoritma Naive Bayes Dalam Memprediksi Tingkat Kelancaran Pembayaran Sewa Teras UMKM’, *Jurnal Informatika*, 8(2), pp. 111–122. Available at: <https://doi.org/10.31294/ji.v8i2.10494>.